The Sound Motion Controller: A Distributed System for Interactive Music Performance

Enrico Cupellini[*1], Jeremy R. Cooperstock[*2], Marta Olivetti Belardinelli[#3]

[*] *Centre for Interdisciplinary Research in Music, Media and Technology, McGill University, Canada*
[#] *Interuniversity Centre for Research on Cognitive Processing in Natural and Artificial Systems (ECONA), Sapienza University, Italy*
[1]enrico.cupellini@gmail.com, [2]jer@cim.mcgill.ca, [3]marta.olivetti@uniroma1.it

ABSTRACT

We developed an interactive system for music performance, able to control sound parameters in a responsive way with respect to the user's movements. This system is conceived as a mobile application, provided with beat tracking and an expressive parameter modulation, interacting with motion sensors and effector units, which are connected to a music output, such as synthesizers or sound effects. We describe the various types of usage of our system and our achievements, aimed to increase the expression of music performance and provide an aid to music interaction. The results obtained outline a first level of integration and foresee future cognitive and technological research related to it.

I. INTRODUCTION

A well-known problem in the design of electronic music devices is giving them the ability to reliably convey human expression to meaningful outputs.

According to Malloch *et al.* (2006) "expressivity" is commonly used to discuss the virtue of an interaction design. An expressive musical interface is considered a device that resonates with its user, creating a better perception of self-awareness, and more accurately achieving the artist's intended sound. The human voice is such an example. It can be adjusted to the artist's intentions, and easily understood by his listeners if they share a common coding with the artist.

In contrast to traditional instruments, electronic music instruments do not provide a direct connection between the artist and the physical sound production mechanism. They tend to express higher levels of separation between the expressive goal of its user and the actual result (Wessel & Wright, 2001). These separations are in part related to the physical material of the interface, which does not give appropriate feedback to the musician. The instrument thus lacks essential elements of communication to the player, and consequently, affects the interaction between the musician and the audience or other players.

We consider a music device as a medium between artist and listener within a social relationship, characterized by attunements and inter-modulations. In this framework, the artist creates symbolic content by playing notes on the instrument, while the system retrieves meaningful information from the artist's movement, and provides feedback by changing values of sound parameters. To support musical expression and interaction between the artist and listener, this process should be effortless, consistent, and in agreement with the musician's intended result.

In the search for more expressive musical interfaces, the sensor-based paradigm appears to better map the performer's actions to musical parameters. Focusing on physically driven interfaces (Tanaka, 2000), we can achieve musical expression employing macro- or micro-levels of movements (Jensenius, 2015), or biological cues related to music production and perception (Muller-Rakow & Fuchs, 2012).

II. APP DESIGN

To support electronic music performance of multiple players, we designed an iOS mobile app called "Sound Motion". The app receives motion data from internal or external sensors, connected by Bluetooth, and drives synthesizers and sound effects as output. To date, we implemented two types of output signal redirection: one using a built-in synthesizer that plays sounds from the same mobile device, and a second that connects the mobile device to a compatible sound module, such as a host or peripheral. For example, OS X 10.10 or higher can use MIDI over Bluetooth and work as a host receiver.

We also prototyped a mechanism to add control of external effects and music hardware, using an Arduino Uno board microcontroller connected to a Bluetooth antenna.

A. User Interface

The app's graphical user interface is shown in Figure 1, with a particular track configuration with three MIDI controls, represented by knobs at the bottom of the screen. Each control can comprise multiple patterns, organized in different pages the user can browse. Each page is represented by a scrollable piano-roll on which the user can insert control events of different length, playing with independent beat subdivisions and independent number of repetitions. Within the app, a track list organizes the control patterns in multiple tracks able to play sequentially. Velocity and other MIDI parameters can be adjusted by the left vertical slider on the piano roll.

During playback, the cursor (gray column) advances, and page turns are automatic.

B. External Modules and Connections

The Sound Motion app receives sensor data either from the internal smartphone accelerometer or from external sensor devices. To date, we have integrated support for a commercial wearable development platform, Hexiwear, that can be used like a smart-watch. Communication between the iOS device and Hexiwear is based on the open-BLE API. Other "open" wearable devices can also be used, provided they support the same Bluetooth protocol.

We also prototyped an effector module to transfer the control patterns directly to analogue music hardware by voltage control changes. This module uses an 8-bit Arduino ATmega microcontroller, with a Bluetooth 4.0 chip (HM-13).

The signal is sent by Bluetooth MIDI from the mobile app to the module, which converts it into voltage changes.

All the signals sent from the mobile app to the external unit employ MIDI protocol so that they can be routed virtually to any sound parameter, managed by the peripheral.

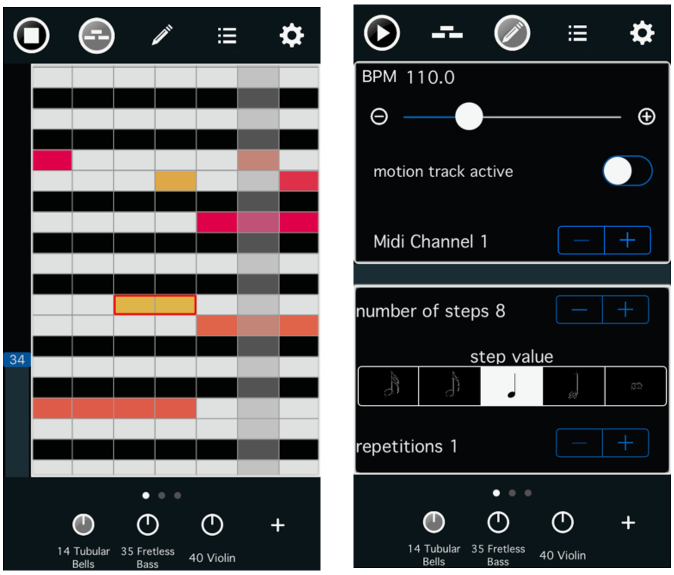

Figure 1. Screenshots of "Sound Motion" mobile app, with the user interface showing the piano-roll page (left) and the editor (right).

III. INTERACTION DESIGN

This section describes the mappings we considered to support musical expression as well as the algorithms used to process the sensor data for modulation of the sound parameters.

A. Direct and Transparent Mappings

Apart from the physical dimension of performance, we must also consider the choice of algorithm that maps the performer's behavior to expressive musical content within an inter-modulation framework. Grosche (2012) for example, claims that given the highly structured level of information in music, which exhibits both time variations and deviation of assumptions, the extraction of musically meaningful information becomes a very challenging problem.

Since acceleration is so intimately involved in music expression, our interaction design focused mainly on the use of accelerometers, worn on the wrist or on the arms of a musician.

For our initial testing, we employed a direct mapping between physical movements and sound output. This followed a gesture-based paradigm that correlates changes of sound amplitude with the performer's physical gestures (Bouillot *et al.,* 2008). The first author used the system in a performance at the Conservatory of Frosinone, Italy, in 2016. The mobile device was tied to an arm and the output control was directed to our hardware prototype, changing the parameters of two pedal effects. However, this approach was limited, because the musician was forced to maintain the same gestures to obtain a given effect. Additionally, any movements, even if unintentional, can result in changes of output.

In our research for a transparent mapping, as promoted by Fels *et al.* (2002), we maintained the metaphor of a sound change related to the amount of motion. However, we separated the creation of control patterns between the symbolic content the composer wants to achieve, and the overlaid expressive content that the performer contributes. To do so, we introduced a sequencer to play music events, which can be programmed to create control patterns, playing together with the musician at a given tempo. Some aspects of these control patterns are influenced by the performer's activity while others remain unvaried, like in a classic music score the written notes are given the particular performer's character.

B. Sonification of Significant Dynamic Ranges

In order to create effective sound expression, we measure the differences of acceleration between local peaks of motion data. From these, we create two dynamic ranges, spanning the high and low ranges of motion, respectively. Figure 2 shows the acceleration changes of a musical gesture and the corresponding ranges of the obtained clusters.

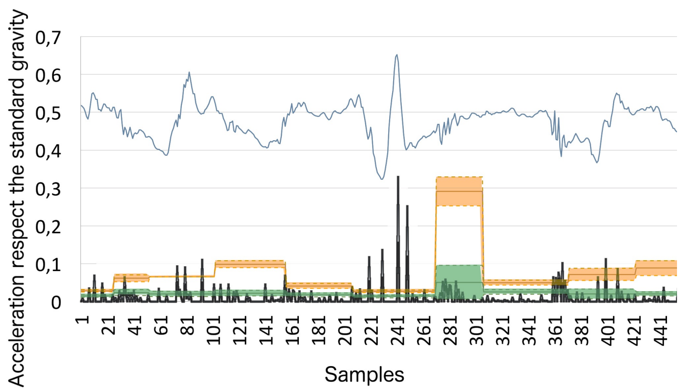

Figure 2. The acceleration magnitudes in three degrees of freedom are summed (upper blue plot), from which the distances between local maxima and minima are calculated (black spikes). Clusters of these distances are then formed (yellow and green) based on the last ten maxima and minima observed. The figure shows also the deviation of each cluster.

We track the accelerometer motion data from all three degrees of freedom ($x_t\, y_t\, z_t$). The input function i results as the sum of their modulus, $i = |x_t| + |y_t| + |z_t|$, and is thus independent of the sensor orientation. The algorithm measures the distances, d_i, between successive maxima and minima of the input function (in Figure 2 they are represented by the black spikes). Greater distances correspond to greater differences in peak accelerations and decelerations of the performer. The *k-means* algorithm is then applied to the set of distances, $D = \{d_1, \dots, d_N\}$, to form two clusters, $S_1 = \{d_k : |d_k - \mu_1| \leq |d_k - \mu_2|\}$, $S_2 = \{d_k : |d_k - \mu_2| \leq |d_k - \mu_1|\}$, corresponding to small and large distances between the peaks.

The minimum and maximum samples of the last ten elements of D are used as initial centroid values for the *k-means* algorithm. From a dynamic point of view, the lower cluster contains less significant actions and therefore is used like a noise threshold, on the other side, we consider the range of $[\mu_1 + \sigma_1, \mu_2 + \sigma_2]$ as significant for dynamic accents and we exploit it in order to map musical expressive gestures to sound. These gestures are mapped as perceptive accents to the corresponding parameter value being controlled, for example, velocity, pitch-shift, after-touch or other continuous controls. We found that the amount of expression m can be mapped according to the equation:

$$m = J \arctan\left[K \frac{d_i - (\mu_1 + \sigma_1)}{\mu_2 + \sigma_2 - (\mu_1 + \sigma_1)}\right] + Q$$

where J is related to the amplitude of modulation, K affects the amplification ratio between softer and stronger values and Q translates the modulation to a desired auditory domain. Figure 3 represents the mapping result by vertical spikes, using $J=0.68$ and $K=1.95$. While in Figure 3A $Q=0$, in Figure 3B $Q=-0.5$. These two solutions generate different auditory effects, in fact, while the first conveys significant gestures to positive accents, the second solution gives order to the gestures as negative and positive accents, according to the amount of acceleration. The *arctan* function is used to compress spikes resulting from differences in acceleration exceeding our range of interest.

Since the cluster parameters are adapted continuously throughout the performance, smaller gestures are still recognized and interpreted as musically meaningful, provided they do not occur within or immediately after a set of larger ones. To ensure smooth size changes in our range of interest, we interpolate the values over a queue made by the last three clustering results. The interpolated range boundaries are shown in Figure 3 by the black segmented lines.

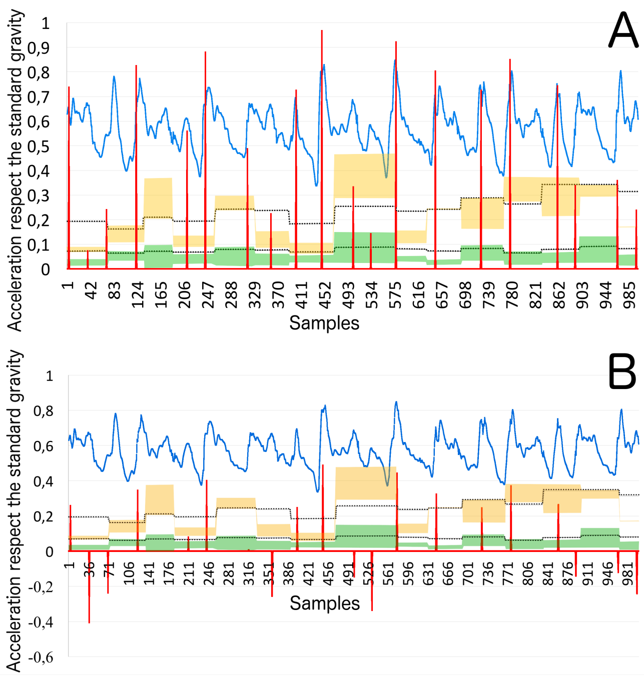

Figure 3. The expressive algorithm calculates the distance between successive inflection points of the acceleration function, compare them with a range of past differences and assigns an expression value, represented by the red vertical spikes. These spikes are associated with modulations of musical parameters. The boundaries of the range of interest are represented by the dashed lines in the bottom. Figure 3A describes the mapping of significant accelerations over positive auditory accents, while Figure 3B represents a mapping over positive and negative accents.

C. Beat Tracking

Tempo and beat retrieval are of fundamental importance to understanding and interacting with music. Musical time is hierarchically structured on a beat and its subdivisions. Several musical components such as pitch, dynamics, timbre, duration are also structured at different levels and can create recurrences, often over beat multiples. Their variation is expressively employed at different levels of importance to create a dynamic attending to specific events in time (Brunetti *et al.,* 2005). As claim Jones & Boltz (1989) this attending behavior involves entrainment to a referent time period and selective focal attending to some time level.

We can find in literature works discussing the participated tuning of timing and dynamics during the music performance. For example, Jensen *et al.* (2010) say that especially in small ensembles, musicians has to depend on themselves each other, as a whole. They distinguish between symmetric and complementary behavioral patterns and they describe proxemic and kinetic cues modulated during the music performance.

Timing is managed in Sound Motion by a beat-tracking algorithm, which employs DBSCAN and a categorization of motion samples. This approach is inspired by the *beat error histogram* method (Davies *et al.,* 2011), it considers the fractions of a given beat duration and compares these to the time intervals between relevant acceleration changes made by the performer.

Since we are interested in conditions in which musicians play with a steady beat, unconstrained by a metronome, we need an algorithm that tracks a natural modulation of tempo and returns stable results, without abrupt changes. Autocorrelation is a good method for this purpose, but it requires rhythmic and periodic signals, which are not always available from arm motion.

Instead, we use the same motion peaks described above, extracting the inter-time interval (ITI) between consecutive expressive triggers, as shown in Figure 3. We start a new BPM detection when at least ten triggers have been collected in a time period greater than 2 s from the last evaluation. To calculate BPM, a DBSCAN algorithm extracts the most representative periods from the last collection of ITI durations. The mean of each cluster is compared to a grid of beat subdivisions, such as the one shown in Figure 4, to assess the error between the exact and real values, to update the BPM value as appropriate.

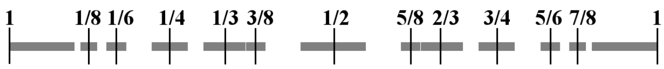

Figure 4. Beat subdivisions and related time regions of error around each subdivision of one cycle (a beat). The Inter-Time Intervals (ITI) mean values of each cluster are matched to this grid of beat subdivisions to assess whether a given rhythmical category can be recognized. The grid is used recurrently for multiple beat durations.

We used the DBSCAN algorithm because we do not know, a priori, what subdivision the musicians will play by their own movements. The algorithm creates clusters with at minimum two elements and a neighbor distance equal to one eighth of the beat duration. Only those clusters with low standard deviation and with the mean value occurring within an error class are considered for beat tracking. The allowed error region for each beat fraction category is calculated to avoid overlap and to be greater than 0.2 s. Thus, fewer categories are feasible for the higher BPM rates than for the lower ones.

IV. EXPERIMENTAL RESULTS

The data in Figures 2 and 3, spanning respectively a time of approximately 4 and 9 s, represent the acceleration values recorded during a guitar session. The guitarist played with a strumming technique, while wearing the accelerometer on the right wrist. The clustering process illustrated in these figures demonstrates the output of the expressive algorithm in Figure 3 from vertical spikes, which result from the processing of local acceleration peaks. The expression can be conveyed in different ways to confer an accent to the next control event the sequencer will play. The events are played cyclically, in loop by the app, the user interacts on each recurrence to confirm or disconfirm a given expectation.

We tested the system in a live music session with a guitarist, a keyboardist, and a drummer. Each musician used our system individually, with the mobile app connected to the external wearable sensor. Since we used the Hexiwear firmware in its default factory settings, which offered a relatively low update rate of accelerometer values, the test involved only the use of the expressive control algorithm, and not the beat tracking.

Each musician, in turn, held their sensor on the wrist of their playing hand and was asked to set up the Bluetooth connection between the Hexiwear sensor and the output device, a MacBook Pro that played a virtual instrument synthesizer in accompaniment. They each created a pattern of control events, and then played together with the Sound Motion system in a five-minute group performance.

Following the performance, the musicians answered a short questionnaire about their musical impressions, ease of use of the system, and subjective physical constraints. Despite initial difficulty in establishing connections and drawing the music events on the piano roll, all participants reported positive feedback regarding their musical impressions of the system. Once the pattern was playing, the interaction with their movements was highly appreciated with no perceived physical constraint.

In order to test the beat tracking algorithm, we used a different approach. We recorded the beat times from a metronome along with the events played by the mobile app sequencer. At the beginning of each test, the BPM of the mobile app was set equal to that of the external metronome. During the test, the metronome changed its rate by +/- 8 BPM each minute, while the musician played the drums in beat with the metronome, unaware of the beat-tracking results. The musician wore the mobile device on his wrist and played different rhythms in turn. For this test, we used the built-in accelerometer of the mobile device, since the Hexiwear default update rate was too slow for timing evaluation.

Figure 5 shows the results over a time span of 11 minutes. The metronome values, represented by the blue line, varied between 100 and 140 BPM; corresponding to beat durations between 600 and 428 ms.

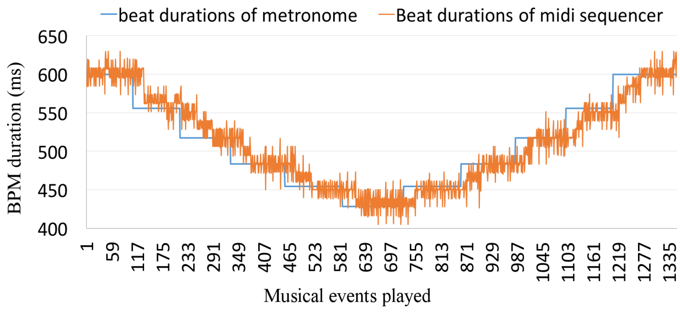

Figure 5. Beat tracking algorithm performance. The blue line describes the durations of the metronome from 100 BPM to 140 BPM and back to 100 BPM. The orange line describes the note durations played by the mobile app sequencer during the test, resulting from the beat tracking algorithm.

V. DISCUSSION

The expressive algorithm proved effective at tracking changes in the performer's movements, and added coherent musical accents to the pattern values played by the sequencer. When the accelerometer is properly worn, these accents can convey the musician's expression, enriching the listener's experience. We would like to dwell on the different results illustrated in Figure 3. The mapping used in Figure 3A which delivers as its output only positive accents, can be considered congruent to map the expression of the higher dynamic range. Though musicians tend to play louder respect than when they use the expressive algorithm as described in Figure 3B. The mapping which delivers positive and negative accents was related in our observations to more balanced dynamics. Perhaps this was caused because in the first case the auditory domain of sonification is not well understood by the musicians who don't find an appropriate mapping for their softer/significant gestures. In the second case instead, they are aware of the possibility of subtle expression through the system.

The results obtained by the beat tracking algorithm are shown in Figure 5. The durations played by the mobile app follow the metronome changes with some slight timing errors around stable durations. This is due to computing issues related to the Bluetooth connection and internal processing. However, the resulting output value did not fluctuate. Each new BPM evaluation is requested every ten triggers and the final output depends on the clusters formed from the ITI population, provided that they occur within a beat fraction category; otherwise the BPM is not changed. The grid of beat fractions against which the measured beat is compared has some discontinuities, as can be seen in Figure 4. This implies that the musician must play precisely over several measures to change the beat value.

We measured a noticeable delay between the environmental metronome changes and the beat tracking response. This delay is likely due to the considerable magnitude of metronome changes during the test, whereas our algorithm is designed for slow beat changes.

The Bluetooth MIDI connection between our hardware and the computer was found to be reliable during the tests. We could play virtual instruments and record experimental data with no noticeable interruptions.

With respect to the external sensor, the general pairing procedure should allow our system to be used by different wearable devices. Although the particular device we tested proved suitable for our expressive control algorithm, its factory-default update rate was too slow for beat tracking. Thus, in future tests, we should update its firmware or consider other wearable devices.

VI. CONCLUSIONS AND FUTURE WORK

To summarize, the results we obtained confirm the expectations for a mobile application delivering interactive music control by translating motion-related cues into expressive musical parameter modulations. The system aims to be a flexible and distributed tool for music performance; affordable, non-intrusive, and effective in practical use. For this purpose, we aimed to support a variety of use cases and connections between sensors and music output units.

We proposed specific algorithms to address the interaction with the musician's behavior. These algorithms were inspired by our background in music perception and experience in behavioral analysis. For instance, our beat tracking works by processing motion sensor values and comparing the resulting times to categories of beat fractions, more or less fractioned according to the BPM rate. This beat tracking is not designed to detect the BPM in a song but to follow the slight human timing changes during a performance, and allow musicians to "push" or "pull" the beat duration to effect some coherent change. Future work will evaluate whether musicians playing along with Sound Motion take advantage of this capability.

The expressive algorithm has been designed to highlight a significant range of motion that changes dynamically according to the performer's expression, and can comprise macro- or micro-levels of motion. We have noticed different interactions regard the mapping used. A potential improvement to this algorithm would be to consider conditions under which to stop triggering the expressive and beat tracking algorithms, for example, when the user has not made any significant gestures for a specific period of time. This would help avoid the generation of unintentional outputs.

With respect to our goal of supporting expressive control and improved interaction between musicians, our future work aims to allow connections and interactions between different mobile devices. We will carry out testing in interplay conditions, working to improve the algorithms and the usability of the system. As one example of such improvements, we are planning to implement some configuration presets, especially for the Bluetooth connection, and introduce a simplified control mechanism to simplify the learning curve for novice users. This was prompted by comments received as feedback during our initial testing.

Additionally, the hardware prototype will be improved to receive and send commands to the mobile device, and tested on stage in a future performance. Finally, we hope that the system's flexibility will encourage users to experiment by connecting other sensors.

ACKNOWLEDGMENT

We would like to acknowledge the musicians who tested and gave suggestions to the work, and in particular the producer Ernesto Ranieri for the joined work of ideation, the music hardware manufacturer Massimo Pastori for the help in making our output module prototype, the guitarist Bruno Marrazzo for the invaluable teaching and support, the test sessions and the performance collaboration at the Conservatory of Frosinone, Italy. This paper is dedicated to Francesco Rosato (1977-2016), dear friend and amazing guitarist.

REFERENCES

Brunetti, R., Cupellini, E., & Olivetti Belardinelli, M. (2005). Entrainment to isochronous and non-isochronous meters: A comparison between tapping by musicians and non-musicians. In *ESCoP 14th: Proceedings of the 14th International Conference of European Society for Cognitive Psychology*. Leiden, Nederland.

Buillot, N., Wozniewski, M., Settel, Z., & Cooperstock, J. R. (2008). A Mobile Wireless Augmented Guitar. In *NIME '08: Proceedings of the 8th International Conference on New Interfaces for Musical Expression.* Genova, Italy.

Davies, M. E. P., Negara, N., & Plumbley, M. D. (2011). Measuring the performance of beat tracking algorithms using a beat error histogram. *IEEE Signal Processing Letters, 18*(3), 157-160.

Fels, S., Gadd, A., & Mulder, A. (2002). Mapping transparency through metaphor: Towards more expressive musical instruments. *Organised Sound*, *7*(2), 109-126.

Grosche, P. M. (2012). *Signal Processing Methods for Beat Tracking, Music Segmentation, and Audio Retrieval.* Doctoral dissertation, Universität des Saarlandes, Germany.

Jensen, K. K., & Marchetti, E. (2010). A Meta-study of musician's non-verbal interaction. *The International Journal of Technology Knowledge & Society, 6*(5), 1-12.

Jensenius, A. R. (2015). Microinteraction in music/dance performance. In *NIME '15: Proceedings of the International Conference on New Interfaces for Musical Epression.* Baton Rouge, LA: Louisiana State University.

Jones, M. R., & Boltz, M. (1989). Dynamic attending and responses to time. *Psychological Review*, *96*(3), 459-491.

Malloch, J., Birnbaum, D., Sinyor, E., & Wanderley, M. M. (2006). Towards a new conceptual framework for digital musical instruments. In *DAFx-06: Proceedings of the 9th International Conference on Digital Audio Effects (*pp. 49-52). Montreal, Canada.

Muller-Rakow, A., & Fuchs, J. (2012). The Human Skin as an Interface for Musical Expression. In *NIME '12: Proceedings of the International Conference on New Interfaces for Musical Expression.* Ann Arbor, Michigan. University of Michigan.

Tanaka, A. (2000). Musical performance practice on xensor-based instruments. In M. M. Wanderley & M. Battier (Eds.), *Trends in Gestural Control of Music.* Ircam - Centre Pompidou.

Wessel, D. & Wright, M. (2001). Problems and prospects for intimate musical control of computers. In *NIME '01: Proceedings of the International Conference on New Interfaces for Musical Expression,* Seattle, WA. New York, NY: ACM Press.